WML Reference Guide

Write nice WAP code.

<a>

Specifies that the text within the tags a hyperlink. The destination of a link is specified as a URI: the address or ID of another tag. Authors are encouraged to use the a tag instead of anchor where possible. It is invalid to nest anchor tags. The a tag is a short form of the anchor tag; it is essentially bound to a go task without variables.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
href	HREF	●		The natural or formal language of the tag or its attributes
title	String			Specifies a brief text string identifying the link

<access>

Specifies access control information for the entire deck. It is an error for a deck to contain more than one access element. If a deck does not include an access element, access control is disabled. When access control is disabled, cards in any deck can access this deck.

A deck's domain and path attributes specify which other decks may access it. As the browser navigates from one deck to another, it performs access control checks to determine whether the destination deck allows access from the current deck.

If a deck has a domain and/or path attribute, the referring deck's URI must match the values of the attributes. Matching is done as follows: the access domain is suffix-matched against the domain name portion of the referring URI and the access path is prefix matched against the path portion of the referring URI.

Domain suffix matching is done using the entire element of each sub-domain and must match each element exactly (e.g. www.wapforum.org shall match wapforum.org, but shall not match forum.org).

Path prefix matching is done using entire path elements and must match each element exactly (e.g. /X/Y matches /X in the attribute, but does not match /XZ in the attribute).

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
domain	String		Current deck's domain	The domain suffix of allowed referring pages.
path	String		/	The path prefix of allowed referring pages.

Nested tags:
 <img>

<anchor>

Specifies that the text within the tags a hyperlink. The destination of a link is specified as a URI: the address or ID of another tag. Authors are encouraged to use the a tag instead of anchor where possible. It is invalid to nest anchor tags.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
title	String			Specifies a brief text string identifying the link

Nested tags:
 <img> <go> <prev> <refresh>

<b>

Indicates that the text within the tags should be rendered with bold formatting. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <b> <i> <u> <big> <small>

 <img> <anchor> <a> <table>

<big>

Indicates that the text within the tags should be rendered with a large font.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Ends the current line and starts a new line. Should also be supported within tables.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

<card>

A WML deck contains a collection of cards. The card element is a container of text and input elements that is sufficiently flexible to allow presentation and layout in a wide variety of devices, with a wide variety of display and input characteristics. A card can contain markup, input fields and elements indicating the structure of the card.

A card's id may be used as a fragment anchor.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
xml:lang	String			The natural or formal language of the tag or its attributes
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
title	String			Specifies advisory information about the card. The title may be rendered in a variety of ways by the browser (suggested bookmark name, pop-up tooltip, etc.)
newcontext	Boolean		FALSE	Indicates that the current browser context should be re-initialised upon entry to this card
ordered	Boolean		TRUE	Specifies a hint to the browser about the organisation of the card content. May be used to organise the content presentation or to otherwise influence layout of the card
onenterforward	HREF			The URI that is loaded when the user navigates into the card using a go task
onenterbackward	HREF			The URI that is loaded when the user navigates into the card using a go task
ontimer	HREF			The URI that is loaded when the timer expires

Nested tags: <do> <onevent> <timer> <p>

<do>

The do tag provides a general mechanism for the user to act upon the current card. The representation of the do tag is dependent on the device and the author must only assume that the tag is mapped to a unique user interface widget, such as a button, that the user can activate.
The do tag may appear at both the card and deck-level:

* Card-level: the do tag may appear inside a card tag and may be located anywhere in the text flow.

* Deck-level: the do tag may appear inside a deck template, indicating a deck-level do tag. A deck-level do tag applies to all cards in the deck, and is equivalent to having specified the do within each card. For the purposes of rendering, the browser must behave as if deck-level do tags are located at the end of the card's text flow.

A card-level do tag overrides (or "shadows") a deck-level do tag if they have the same name. For a single card, the active do tags are defined as the do tags specified in the card, plus any do tags specified in the deck's template and not overridden in the card.

Non-active do tags and active do tags with a noop task are not displayed. All active do tags with a task other than noop will be shown in some manner. When the user activates the do tag, the associated task is executed.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
xml:lang	String			The natural or formal language of the tag or its attributes
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
type	String	●		Provides a hint to the browser about the author's intended use of the tag and how the tag should be mapped to the physical user interface: 'accept' (positive acknowledgement); 'prev' (backward history navigation); 'help' (request for help); 'reset' (clearing or resetting state); 'options' (request for options or additional operations); 'delete' (delete item or choice); 'unknown' a generic do element
label	String			Specifies a textual string suitable for labelling the user widget. To work well on a variety of browsers, labels should be six characters or shorter in length
name	String		Value of type attribute	Specifies the name of the do event binding
optional	Boolean		FALSE	If true, the browser may ignore this element

Nested tags: <go> <prev> <noop> <refresh>

Indicates that the text within the tags should be rendered with some form of emphasis. Authors should attempt to use the em and strong tags in place of the b, i and u tags, except where explicit control over text presentation is required.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table>

<fieldset>

The fieldset element allows the grouping of related fields and text. This grouping allows the optimising of layout and navigation. Fieldset elements may nest, providing the author with a means of specifying behaviour across a wide variety of devices

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
title	String			Specifies a title for this element, which may be used in the presentation of this object.

Nested tags: <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table> <input> <select> <fieldset> <do>

<go>

Declares a 'go' task, indicating navigation to a new URI. If the URI names a WML card or deck, the execution of the task will cause that item to be displayed. This task executes a 'push' operation on the browser's history stack.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
href	HREF	●		Specifies the destination URI
sendreferer	Boolean		FALSE	If true, the request for the new URI will contain the URI of the deck containing this task - this can then be used for server-based access control
method	METHOD		GET	Specifies the HTTP submission method. Possible values are 'POST' (submission data included in request) or 'GET' (submission data appended as a query to the URI requested)
accept-charset	String		unkown	Specifies the list of character encodings for data that the origin server must accept when processing input. The value of this attribute is a comma- or space-separated list of character encoding names

Nested tags: <postfield> <setvar>

<head>

The head tag contains information relating to the deck as a whole, including meta-data and access control tags.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

Nested tags: <access> <meta>

<i>

Indicates that the text within the tags should be rendered with italic formatting. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table>

The img tag indicates that an image is to be included in the text flow. Image layout is done within the context of normal text layout.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
alt	String	●		An alternative textual representation for the image if it can not be displayed using any other method
src	URI			This attribute specifies the URI of the image to be shown
localsrc	String			Specifies an alternative internal representation for the image. If this image exists, the parameter takes precedence over any image specified in the src parameter
vspace	Length	●	Small, non-zero length	Specifies the amount of white space to be inserted above and below the image. If specified as a percentage value, the resulting space is based on the available vertical space, not on the natural size of the image
hspace	Length	●	Small, non-zero length	Specifies the amount of white space to be inserted to the left and right of the image. If specified as a percentage value, the resulting space is based on the available horizontal space, not on the natural size of the image
align	String	●	Bottom	Specifies image alignment with respect to the baseline of the text. It has three possible values: ‘bottom’, ‘middle’ or ‘top’
height	Length	●		Specifies the height of the image. If specified as a percentage value, the resulting size is based on the available vertical space, not on the natural size of the image
width	Length	●		Specifies the width of the image. If specified as a percentage value, the resulting size is based on the available horizontal space, not on the natural size of the image

<input>

The input element specifies a text entry object. The user input is constrained by the optional format attribute.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
name	String	●		Specifies the name of the variable to set with the result of the user's text input. The name variable's value is used to pre-load the text entry object. If this value does not conform to the input mask, the browser will unset the variable and attempt to initialise the variable with the value attribute
type	INPUT_TYPE		text	Specifies the type of text-input area. The following values are allowed: 'text' (text entry control) or 'password' (a text entry control where each character is displayed in an illegible form, such as with asterisks)
value	String			The value attribute indicates the default value of the variable named in the name attribute. When the element is displayed and the variable named in the name attribute is not set, the name variable is assigned the value specified in the value attribute. If the name variable already contains a value, the value attribute is ignored. If the value attribute specifies a value that does not conform to the input mask specified by the format attribute, the browser must ignore the value attribute. In the case where no valid value can be established, the name variable is left unset
format	String		*M	Specifies an input mask for user input entries. The string consists of mask control characters and static text that is displayed in the input area. The format control characters specify the data format expected to be entered by the user. The format codes are: 'A' (entry of any upper-case, non-numeric character); 'a' (entry of any lower-case, non-numeric character); 'N' (entry of any numeric character); 'X' (entry of any upper-case character); 'x' (entry of any lower-case character); 'M' (entry of any character); 'm' (entry of any character); '*f' (entry of any number of characters where f is one of the above format codes); 'nf' (entry of n characters where n is from 1 to 9 and f is one of the above format codes); '\c' (display the character, c, in the entry field: these characters are considered part of the input's value).
emptyok	Boolean		False	Indicates that this input element accepts empty input although a non-empty format string has been specified

size	NUMBER			Specifies the width, in characters, of the text-input area.
maxlength	NUMBER		Infinite	Specifies the maximum number of characters that can be entered by the user in the text-entry area.
tabindex	NUMBER			Specifies the tabbing position of the current element. The tabbing position indicates the relative order in which elements are traversed when tabbing within a single WML card.
title	String			Specifies a title for this element, which may be used in the presentation of this object.

<meta>

The meta element contains generic meta-information relating to the WML deck. Meta-information is specified with property names and values. This specification does not define any properties, nor does it define how browsers must interpret meta-data.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
http-equiv	String			May be used in place of name and indicates that the property should be interpreted as an HTTP header.
name	String			Specifies the property name. The browser must ignore any meta-data named with this attribute. Servers should not emit WML content containing meta-data named with this attribute.
forua	Boolean			Specifies that the author intended the property to reach the browser. In the case where the user agent supports the meta-data mechanism, and the property has its forua attribute set to true, the meta-data must be delivered to the browser.
content	String	●		This attribute specifies the property value.
scheme	String			This attribute specifies a form or structure that may be used to interpret the property value. Scheme values vary depending on the type of meta-data.

<noop>

The noop tag specifies that nothing should be done - that is, 'no operation'. This can be used in a card to shadow a task that has been specified in a template at the deck level.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

<onevent>

The onevent element binds a task to a particular intrinsic event for the immediately enclosing element. For example, specifying an onevent element inside a card element associates an intrinsic event binding with that card element. The browser will ignore any onevent element specifying a type that does not correspond to a legal intrinsic event for the immediately enclosing element.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
type	String	●		Indicates the name of the intrinsic event

Nested tags: <go> <prev> <noop> <refresh>

<optgroup>

The optgroup element allows the author to group related option elements into a hierarchy. The browser may use this hierarchy to facilitate layout and presentation on a wide variety of devices.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
title	String			Specifies a title for this element, which may be used in the presentation of this object.

Nested tags: <optgroup> <option>

<option>

This element specifies a single choice option in a select element.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
value	String			Specifies the value to be used when setting the name variable. When the user selects this option, the resulting value specified in the value attribute is used to set the select element's name variable. The value attribute may contain variable references, which are evaluated before the name variable is set.
title	String			Specifies a title for this element, which may be used in the presentation of this object.
onpick	HREF			The URI that is loaded when the user selects or deselects this option.

Nested tags: <onevent>

<p>

The p element establishes both the line wrap and alignment parameters for a paragraph. If the text alignment is not specified, it defaults to left. If the line-wrap mode is not specified, it is identical to the line-wrap mode of the previous paragraph in the current card. Empty paragraphs (ie, an empty element or an element with only insignificant white space) will be considered as insignificant and ignored by browsers. If the first p element in a card does not specify a line-wrap or alignment mode, that mode defaults to the initial mode for the card.

The browser will insert a line break into the text flow between significant paragraphs.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
align	TALIGN		left	Specifies the text alignment mode for the paragraph. Can be 'left', 'center', or 'right'.
mode	WRAPMODE		wrap (or as in previous paragraph)	Specifies the line-wrap mode for the paragraph. 'wrap' specifies breaking text mode and 'nowrap' specifies non-breaking text mode.
xml:lang	String			The natural or formal language of the tag or its attributes
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

Nested tags: <do>   <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table> <input> <select> <fieldset>

<postfield>

The postfield element specifies a field name and value for transmission to an origin server during a URL request. The actual encoding of the name and value will depend on the method used to communicate with the origin server.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
name	String	●		Specifies the variable name.
value	String	●		Specifies the variable name.

<prev>

The prev tag declares a 'prev' task, indicating navigation to the previous URL on the history stack.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

Nested tags: <setvar>

<refresh>

The refresh tag declares a refresh task, indicating an update of the screen and device context as specified by the setvar tags, for example. User-visible side effects of the state changes (e.g. a change in the screen display) occur during the processing of the refresh task.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

Nested tags: <setvar>

<select>

The select element lets users pick from a list of options. Each option is specified by an option element. Each option element may have one line of formatted text. Option elements may be organised into hierarchical groups using the optgroup element.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
title	String			Specifies a title for this element, which may be used in the presentation of this object.
name	String	●		Specifies the name of the variable to set with the result of the selection. The name variable's value is used to pre-load the text entry object.
value	String			Indicates the default value of the variable named in the name attribute. When the element is displayed, and the variable named in the name attribute is not set, the name variable may be assigned the value specified in the value attribute, depending on the values defined in iname and ivalue. If the name variable already contains a value, the value attribute is ignored. Any application of the default value is done before the list is pre-selected with the value of the name variable. If this element allows the selection of multiple options, the result of the user's choice is a list of all selected values, separated by the semicolon character. The name variable is set with this result. Similarly, the value attribute is interpreted as a semicolon-separated list of pre-selected options.
iname	String			Indicates the name of the variable to be set with the index result of the selection. The index result is the position of the currently selected option in the select list. An index of zero indicates that no option is selected. Index numbering begins at one and increases monotonically.
ivalue	String			Indicates the default-selected option element. When the element is displayed, if the variable named in the iname attribute is not set, it is assigned the default-selected entry. If the variable already contains a value, the ivalue attribute is ignored. If the iname attribute is not specified, the ivalue value is applied every time the element is displayed. If this element allows the selection of multiple options, the index result of the user's choice is a list of the indices of all the selected options, separated by the semicolon character. The iname variable is set with this result. Similarly, the ivalue attribute is interpreted as a semicolon-separated list of pre-selected options.
multiple	Boolean		FALSE	Indicates that the select list should accept multiple selections. When not set, the select list should only accept a single selected option.

multiple	Boolean	FALSE	Indicates that the select list should accept multiple selections. When not set, the select list should only accept a single selected option.
tabindex	NUMBER		Specifies the tabbing position of the current element. The tabbing position indicates the relative order in which elements are traversed when tabbing within a single WML card.
xml:lang	String		The natural or formal language of the tag or its attributes

Nested tags: <optgroup> <option>

<setvar>

The setvar element specifies the variable to set in the current browser context as a side effect of executing a task. The element must be ignored if the name attribute doe not evaluate to a legal variable name at runtime.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
name	String	●		Specifies the variable name.
value	String	●		Specifies the value to be assigned to the variable.

<small>

Indicates that the text within the tags should be rendered with a small font.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table>

Indicates that the text within the tags should be rendered with some form of strong emphasis. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table>

<table>

Used together with the tr and td tags to create sets of aligned columns of text and images in a card. The table tags determine the structure of the columns. The tags separate content into columns, but do not specify column or intercolumn widths.

The number of columns for the row set must be specified by the columns attribute. If the actual number of columns in a row is less than the value specified by the columns attribute, the row will be effectively padded with empty columns. The orientation of the table depends on the language. For left-to-right languages, the leftmost column is the first column in the table. Columns are added to the right side of a row to pad left-to-right tables. Columns are added to the left side of a row to pad right-to-left table.

If the actual number of columns in a row is greater than the value specified by the columns attribute, the extra columns of the row will be aggregated into the last column, such that the row contains exactly the number of columns specified. A single inter-word space will be inserted between two cells that are being aggregated.

The table will be rendered as narrow as possible given the contents. A non-zero width gutter is used to separate each non-empty column.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes
title	String			Specifies a brief text string identifying the table
align	String		Left	Specifies the layout of text and images (within the columns) as a list of alignment designations - one for each column. Centre alignment is specified with C, left alignment with L, and right alignment with R. If an alignment designation is omitted, the default alignment is applied
columns	NUMBER	●		Specifies the number of columns in the table. Must not be zero

Nested tags: <tr>

<td>

The td element is used as a container to hold a single table cell data within a table row. Table data cells may be empty. Empty cells are significant, and must not be ignored. The browser will attempt to deal with multiple line data cells that may result from using images or line breaks.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <img> <anchor> <a>   <b> <i> <u> <big> <small>

<template>

The template element declares a template for cards in the deck. Event bindings specified in the template element (e.g. do or onevent) apply to all cards in that deck, although a card element may override the behaviour specified in the template element. In particular:

* DO elements specified in the template element may be overridden in individual cards if both elements have the same NAME attribute value.

* Intrinsic event bindings specified in the template element may be overridden by the specification of an event binding in a card element.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
onenterforward	HREF			The URI that is loaded when the user navigates into a card using a go task

onenterbackward	HREF	The URI that is loaded when the user navigates back into a card using a prev task
ontimer	HREF	The URI that is loaded when the timer expires

Nested tags: <do> <onevent>

<timer>

The timer element declares a card timer which exposes a means of processing inactivity or idle time. The timer is initialised and started at card entry and is stopped when the card is exited. Card entry is any task or user action that results in the card being activated, for example, navigating into the card. Card exit is defined as the execution of any task. The value of a timer will decrement from the initial value, triggering the delivery of an ontimer intrinsic event when it reaches zero. Note that timer resolution and the interaction of the timer with the browser's user interface and other time-based device functionality is implementation dependent.

It is an error to have more than one timer element in a card.

The timer timeout value is specified in units of one-tenth (1/10) of a second. The author should not expect a particular timer resolution and should provide the user with another means to invoke a timer's task. A timeout value of zero disables the timer.

Invoking a refresh task is considered an exit. The task stops the timer, commits its value to the context, and updates the browser accordingly. Completion of the refresh task is considered an entry to the card. At that time, the timer must resume.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
name	String			Specifies the name of the variable to be set with the value of the timer. The name variable's value is used to set the timeout period upon timer initialisation. The variable named by the name attribute will be set with the current timer value when the card is exited or when the timer expires. For example, if the timer expires, the named variable is set to a value of zero.
value	String	●		Indicates the default value of the variable named in the name attribute - that is the timer value. When the timer is initialised and the variable named in the name attribute is not set, the name variable is assigned the value specified in the value attribute. If the name variable already contains a value, the value attribute is ignored. If the name attribute is not specified, the timeout is always initialised to the value specified in the value attribute.

<tr>

The tr element is used as a container to hold a single table row. Table rows may be empty (i.e., all cells are empty). Empty table rows are significant and must not be ignored.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

Nested tags: <td>

<u>

Indicates that the text within the tags should be rendered with underline formatting. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member
xml:lang	String			The natural or formal language of the tag or its attributes

Nested tags: <b> <i> <u> <big> <small>
 <img> <anchor> <a> <table>

<wml>

The wml tag defines a deck and encloses all information and cards in the deck.

Attributes:

Name:	Data	Mandatory:	Default:	Description:
xml:lang	String			The natural or formal language of the tag or its attributes
id	String			A unique name for the tag within the deck
class	String			A name of a class of which the tag is a member

Nested tags: <head> <template> <card>